Q2) Code

**Product.cs**

public class Product : IComparable<Product>

{

public int ProductId { get; set; }

public string ProductName { get; set; }

public string Category { get; set; }

public Product(int id, string name, string category)

{

ProductId = id;

ProductName = name;

Category = category;

}

public int CompareTo(Product other)

{

return ProductId.CompareTo(other.ProductId);

}

}

**Program.cs**

using System;

class Program

{

static void Main(string[] args)

{

Product[] productList = new Product[]

{

new Product(101, "Laptop", "Electronics"),

new Product(205, "Shirt", "Clothing"),

new Product(150, "Mobile", "Electronics"),

new Product(303, "Book", "Education")

};

Console.WriteLine("Linear Search:");

var result1 = SearchDemo.LinearSearch(productList, 150);

Console.WriteLine(result1 != null ? $"Found: {result1.ProductName}" : "Product not found");

Array.Sort(productList); // Binary search needs sorted array

Console.WriteLine("\nBinary Search:");

var result2 = SearchDemo.BinarySearch(productList, 150);

Console.WriteLine(result2 != null ? $"Found: {result2.ProductName}" : "Product not found");

}

}

**SearchDemo.cs**

public class SearchDemo

{

public static Product LinearSearch(Product[] products, int id)

{

foreach (var product in products)

{

if (product.ProductId == id)

return product;

}

return null;

}

public static Product BinarySearch(Product[] products, int id)

{

int left = 0, right = products.Length - 1;

while (left <= right)

{

int mid = (left + right) / 2;

if (products[mid].ProductId == id)

return products[mid];

else if (products[mid].ProductId < id)

left = mid + 1;

else

right = mid - 1;

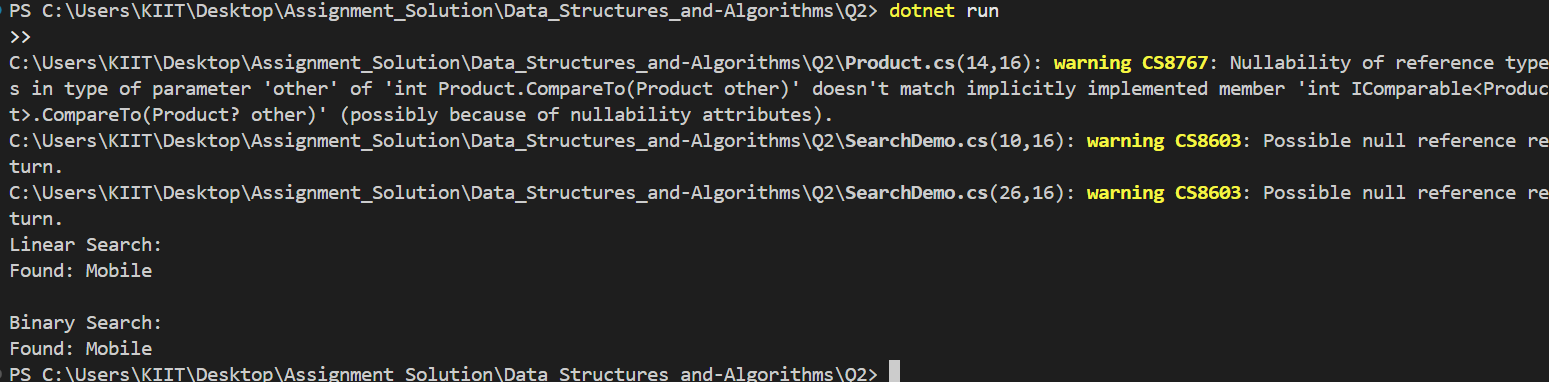
}

return null;

}

}

Output



Q7)

Code

**Forecast.cs**

using System;

public class Forecast

{

// Recursive method to calculate future value

public double CalculateFutureValue(double amount, double rate, int years)

{

if (years == 0)

return amount;

return CalculateFutureValue(amount \* (1 + rate), rate, years - 1);

}

}

**Program.cs**

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter initial amount (e.g., 10000):");

double amount = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Enter annual growth rate in % (e.g., 10 for 10%):");

double rate = Convert.ToDouble(Console.ReadLine()) / 100;

Console.WriteLine("Enter number of years:");

int years = Convert.ToInt32(Console.ReadLine());

Forecast forecast = new Forecast();

double futureValue = forecast.CalculateFutureValue(amount, rate, years);

Console.WriteLine($"Future Value after {years} years = {futureValue:F2}");

}

}

Output

